GUARDIAN: Class/Structure Descriptions

STRUCT GDNcategory ::

Structure that will hold a record found in the category list file (GDN_CategoryList.dat).

Data	TextString/enum?	key	▸	Text key for the category: MON: Meteorological Monitor, SYSTAT: System Status, SYSCOM: System Communications, FORCHK: Forecast Checker, RADAR: Radar messages, TXTPROD: Text Product messages, MISC: Everything else! ☺. Any additional suggestions?
	TextString	name	▸	Full name of the key. (See above.)

STRUCT GDNpriorityInfo ::

Structure that will hold priority-dependent values for each customizable Triggered Action, color, sound, and action..

Data	bool	text	▸	Present text to user.
	bool	blink	▸	Blink text in widget?
	bool	priority	▸	Provide priority in text message?
	bool	popup	▸	Present text to user in red-banner pop-up?
	TextString	background	▸	background color associated to priority. This can be a hex color used by monitor button (begin with A#@), or an image file, or a background color for the text widgets.
	TextString	foreground	▸	foreground color associated to priority. This can be a hex color used by monitor button (begin with A#@) or by the text widgets.
	TextString	audio	▸	sound associated to priority. Can be 0 or 1 for beep on/off, or it can be the name of a sound file.

	TextString	action	▸	action associated to priority

CLASS GDN_ConfigAttribData ::

Contains the data found in the GUARDIAN Config file for one record in the file. Each record in the file uses the "|" character as the delimiter of the listed items. **<u>NOTE</u>**: The _sound, _colors, and _flags vectors should be the same size as they represent what to do for various priority levels!

C'tor Args				none
Data	TextString	_key	▸	user application key (i.e.:"FFMP" or 1001 for FFMP monitor). There will be one key per user application.
	TextString	_gui	▸	tcl script to launch config GUI. - *required for MN type*.
	int	_tInterval	▸	frequency below which notices will **not** be made (in seconds).
	TextString	_bitmap	▸	name of bitmap file (for color-coded button). - *required for MN type*. **(Should this be priority dependent?)**
	TextString/enum ?	_type	▸	type of entry (EV=event, MN=monitor, AC=action). May *drop this item as MN will be the only type with a bitmap list and the only difference between EV and AC is the _action member.*)
	vector<GDNpriorityInfo>	_pInfo	▸	Flags, turning Triggered Actions on/off for each priority level and also other data dependant on priority level, which includes colors, sounds, and actions.
Functions	friend istream &	operator>>()	▸	read in a line of data.
	friend ostream &	operator<<()	▸	write out a line of data
	GDNpriorityInfo &	operator[]()	▸	accessor for indexed items in the '_pInfo' data member
			▸	accessor functions for all private data members

CLASS GDN_Config ::

Contains all of the data found in the GUARDIAN Config file. The constructor will check to see if the file is properly accessible and set the _goodfile flag accordingly. **NOTE:** The config file will have header info that applies to all records (data members of this GDN_Config class - except for the GDN_ConfigAttribData vector), then a listing of the records (the GDN_ConfigAttribData vector). Need to consider whether _log_length actually belongs in the GDN_ConfigAttribData class.

C:tor Args	TextString &		►	file name
Data	vector<GDN_ConfigAttribData>	_entries	►	vector containing data for each record in the config file.
	TextString	_layout	►	layout of the text message lines in the GUI. The format of the layout will be number of lines and symbolic character. Examples:4V, 4Q, 2H, 2V, 3V, 1Vor 1H (V=vertical, H=horizontal, Q=quad). 1V and 1H would mean the same thing.
	vector<GDNcategory>	_layout_list	►	Which categories of text to appear in the layout. Length used can vary depending on _layout chosen.
	int	_log_length	►	Number of messages to keep in each message log
	int	_revert_time	►	time in seconds to revert back to the default configuration - in case something is turned off and forgotten.
	bool	_goodfile	►	Flag marking the file to be accessed as accessible or not.
Functions	bool	read()	►	read in a file of data.
	bool	write()	►	write out a file of data
	bool	open()	►	Actually just checks the value of _goodfile.
	TextString &	getLayoutLMNt(unsigned int)	►	Get an item from _layout_list with the given index.
	GDN_ConfigAttribData &	operator[]	►	accessor for indexed items in the '_entries' data member
			►	accessor functions for all private data members

CLASS GDN_Categories ::

Contains all of the categories that GUARDIAN will recognize. A text file will hold the list and thus can be dynamically altered. MAY NOT NEED THIS CLASS AT ALL (IF WE DON'T USE CATEGORIES AT ALL).

C'tor Args			▸	none, but will read file and fill _categories. If a problem occurs, the vector length of _categories will be zero.
Data	vector<GDNcategory>	_categories	▸	category names, read from a file.
			▸	
Functions	GDNcategory &	operator[]	▸	accessor for indexed items in _categories
		operator>>	▸	to read a record from the file.
	int	size()	▸	the number of items in the vector

CLASS GDN_Container ::

Contains all of the data that the GUARDIAN GUI will be using. Some of this will be passed back and forth to/from the tcl code that is the GUI. Need to decide how best to handle addToLog() and messageReceived() together.

C-tor Args			►	??
Data	GDN_Config	_config	►	Current configuration used by the GUI.
	vector<vector<TextString> >	_messageLogs	►	A list of messages for each Category.
	GDN_Categories	_categories	►	category names, read from a file.
			►	
Functions	bool	addToLog(GDNcategory &, TextString &)	►	When a text message comes in, the text will be added to the message log (and an old message dropped from the log)
	bool	getCategories()	►	read the category file to fill _categories.
		messageReceived(TextString key, TextString mssg, int priority, TextString category)	►	Initiates the chain of events when a message is received by GUARDIAN.
			►	accessor functions for all private data members

Globals (not a class)

Data	small int	GDNPLEVELS	►	Number of different priority levels usable by GUARDIAN. Zero is the HIGHest priority! This way, if the number set gets changed in the future, the 'highest' priority will always be zero
			►	
Functions		gdnNotify()	►	uses enum for category
		gdnNotify()	►	uses TextString for category
			►	

Utility Classes

CLASS	FileFind ::			

Will search for the given file in directories provided upon construction. Can accept environmental variables in the constructor arguments. If the vector argument is empty, the InfoFileServer class is used.

C:tor Args	vector<TextString>		▸	path names
	virtual bool = false		▸	*include* InfoFileServer paths?
Data	vector<TextString>	_paths	▸	A list of messages for each Category.
			▸	
Functions	TextString	getPath(TextString &)	▸	Find the location
			▸	